Приложение В

|  |
| --- |
| **Автономное учреждение  профессионального образования**  **Ханты-Мансийского автономного округа – Югры**  **«СУРГУТСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**  **(АУ «Сургутский политехнический колледж»)** |
|  |
| СТРУКТУРНОЕ ПОДРАЗДЕЛЕНИЕ - 4  (Энергетическое отделение) |

**Отчет по практической работе №\_6\_**

**по дисциплине**

**МДК 01.02 “Инструментальные средства разработки программного обеспечения”.**

Выполнил:

студент группы 319

Тимергалиев Тимур Ильдарович

«\_\_18\_\_» \_\_\_\_11\_\_\_\_\_\_\_\_\_ 2024г.

Сургут – 2024 г.

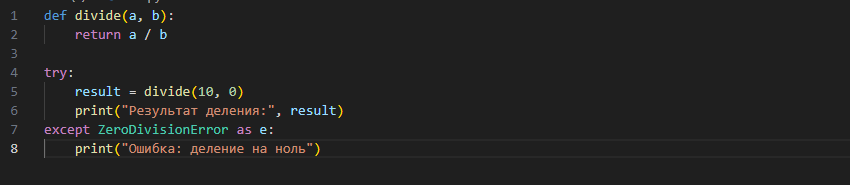
# Цель работы

Цель данной работы – изучить методы и подходы к обработке исключений, научиться применять конструкции try-except, обрабатывать и регистрировать исключения для повышения стабильности программного обеспечения.

# Практическая часть

## Задание 1: Базовая обработка исключений

1. Создайте простой программный модуль, в котором возникает ошибка (например, деление на ноль).



2. Реализуйте обработку исключения с помощью конструкции try-except, чтобы предотвратить аварийное завершение программы.

В приведенном выше коде используется конструкция try-except для обработки исключения ZeroDivisionError.

3. Выведите сообщение об ошибке и продолжите выполнение программы.

При возникновении ошибки программа выведет сообщение "Ошибка: деление на ноль" и продолжит выполнение.

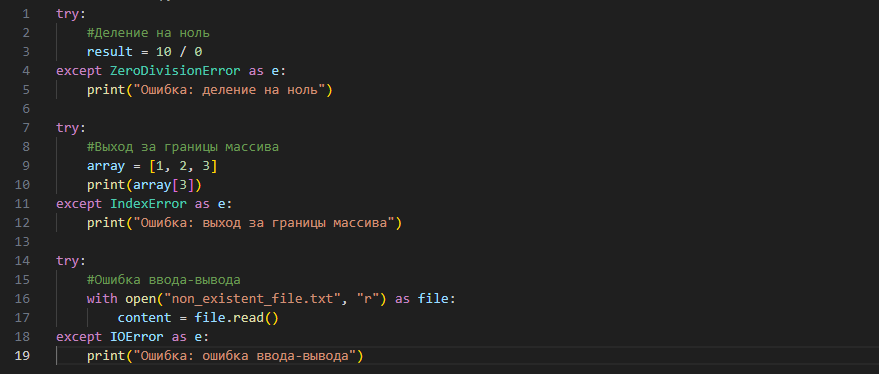
## Задание 2: Иерархия исключений

1. Изучите, как работают различные типы исключений в вашем языке программирования (например, стандартные исключения и их наследники).

В Python существует иерархия исключений, начиная с базового класса BaseException. Некоторые из стандартных исключений:

* ZeroDivisionError: возникает при делении на ноль.
* IndexError: возникает при выходе за границы массива.
* IOError: возникает при ошибках ввода-вывода.

2. Создайте несколько примеров обработки различных исключений (например, деление на ноль, выход за границы массива, ошибки ввода-вывода).

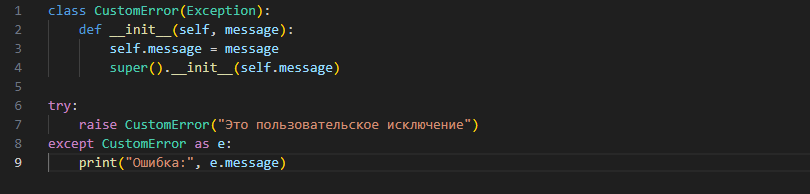


3. Задокументируйте различия между ними и предложите стратегию обработки каждой категории.

* **ZeroDivisionError**: Обрабатывается отдельно, так как это ошибка арифметической операции.
* **IndexError**: Обрабатывается отдельно, так как это ошибка доступа к элементу массива.
* **IOError**: Обрабатывается отдельно, так как это ошибка работы с файлами.

## Задание 3: Генерация пользовательских исключений

1. Создайте пользовательский класс исключений для обработки специфических ситуаций в вашем программном проекте. Напишите пример кода, который генерирует и обрабатывает это исключение.

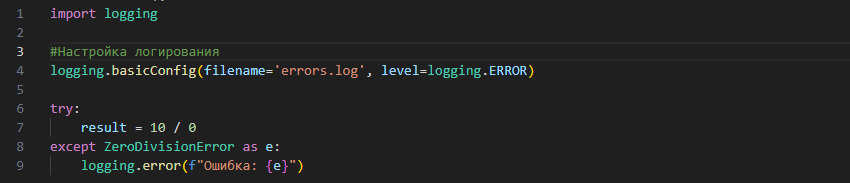


2. Обоснуйте необходимость создания пользовательского исключения.

Пользовательские исключения позволяют обрабатывать специфические ошибки, которые не могут быть охвачены стандартными исключениями. Это улучшает читаемость кода и упрощает отладку.

## Задание 4: Логирование исключений

1. Реализуйте механизм логирования исключений в файл или в базу данных для их дальнейшего анализа.



2. Проверьте корректность логирования и убедитесь, что каждая ошибка записывается с указанием времени, типа ошибки и контекста.

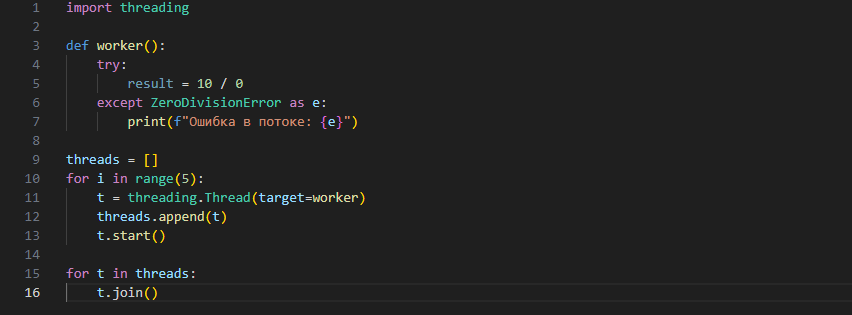
При возникновении ошибки она будет записана в файл errors.log с указанием времени, типа ошибки и контекста.

3. Задокументируйте подходы к анализу и устранению исключений на основе логов.

Анализ логов позволяет выявить часто встречающиеся ошибки и устранить их. Логирование помогает отслеживать состояние программы и быстро находить причины сбоев.

## Задание 5: Обработка исключений в многопоточности

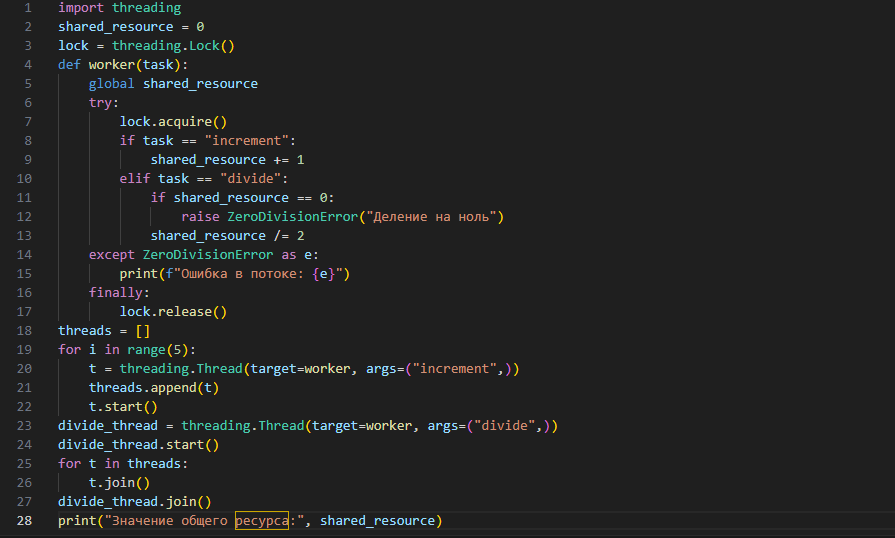
1. Если ваш проект использует многопоточность, реализуйте обработку исключений в многопоточной среде.



2. Убедитесь, что исключения в одном потоке не приводят к завершению программы или других потоков.

В приведенном коде исключение в одном потоке не приводит к завершению программы или других потоков.

3. Используйте синхронизацию для корректной работы с исключениями в многопоточных приложениях.



# Заключение

Я изучил методы и подходы к обработке исключений, научиться применять конструкции try-except, обрабатывать и регистрировать исключения для повышения стабильности программного обеспечения.